Post-Quantum Cryptography: An Overview

Chi Yin Wong 836872

1. Abstract

This paper first provides a brief overview of public key and symmetric encryption schemes that are currently used in practice. Then there is a discussion about quantum computers and two algorithms that they can implement to solve hard mathematical problems at faster speeds, with emphasis on how these mathematical algorithms can weaken or break encryption algorithms. Suggested post quantum cryptographic algorithms will then be explored in detail.

2. Introduction

The advancement of technology and improvement in engineering has resulted in a time in human history where large scale quantum computers are no longer an out of reach dream, but a fast-approaching reality. These powerful computers will be sufficiently strong enough to solve difficult mathematical problems – many of which serve as the bedrock of current public key cryptographic algorithms. This means that although current cryptographic schemes such as RSA and Diffie Hellman are safe against modern technology, there is a motivation to develop more advanced algorithms that are secure enough to withstand the power of a quantum computer. This gives rise to the field of post-quantum cryptography, a field dedicated to exploring more advanced algorithms that are resistant against attacks from quantum computers, as well as improving existing hard mathematical-based schemes, preparing us for an era where quantum computers will be the norm [1].

3. Literature Review

3.1 Current Encryption Schemes

3.1.1 Public Key Encryption Schemes

Many public key cryptographic schemes are based on mathematical problems that are considered hard by today's computing standards. These schemes have been employed for a long period of time and have successfully prevented attacks. One popular scheme is known as RSA, an algorithm used for both key exchange and digital signatures [2]. It is based on two key mathematical problems: integer factorization and finding the eth root mod n. Another popular scheme that is used often for key exchange is the Diffie-Hellman protocol. This protocol is based on the discrete logarithm problem, allowing two parties to communicate by creating a shared secret key without revealing their own private key [3].

3.1.2 Symmetric Key Encryption Schemes

Symmetric key encryption schemes involve the use of a single secret key shared between two communicating parties. The key is used to both encrypt and decrypt messages and the key is assumed to be shared between the parties in advance. The security of these algorithms is based on the assumption that the shared secret key is kept hidden from adversaries. [10] It is often measured by the number of operations it would take to break the algorithm using a brute force attack. Popular symmetric schemes used in practice include the Advanced Encryption Standard (AES) and Triple Data Encryption Standard (3DES) [1].

3.2 Quantum Computers

Quantum computers are machines that employ the full complexity of the properties of quantum physics to store data and perform computations [4]. Rather than encode information in binary bits like today's computers, quantum computers use quantum bits, bits made up of physical systems that can exhibit the property of quantum superposition. Quantum entanglement also allows quantum bits to be linked together [5]. Given a large number of possible combinations such as in hard mathematical problems like integer factorization, quantum computers can consider many computations simultaneously, solving these problems at a much faster rate [6].

3.3 Mathematical Algorithm's Threatening Current Schemes

3.3.1 Shor's Algorithm

Shor's algorithm is a polynomial time quantum algorithm designed to find the prime factorization of any large positive integer n [1]. It has a complexity of $O((\log(n))^2 * \log(log(n)))$ which is much faster than the complexity of classical computers which require exponential time [8]. A quick and very brief overview of Shor's algorithm is as follows. The initial steps of Shor's algorithm include first determining if the integer n is prime, picking an integer q that satisfies $n^2 \leq q < 2n^2$ and picking an integer x that is co-prime to N. Then a quantum computer is used to create a partitioned quantum register with one register having enough quantum bits to represent integers as large as q-1 and the other to represent n-1. Register one will be loaded with an equally weighted superposition of integers from 0 to q-1 and register 2 will be loaded in the 0 state. A transformation of $x^a mod\ n$ (where a is the superposition of the states) is applied to every integer stored in register one and the result is stored in register two. A discrete Fourier transform is computed on register one and a value m which is the state of register one will have a very high probability of being a multiple of q/r where r is unknown and can be found with post processing techniques on classical computers based on knowing m and q. A factor of n can be found by taking the greatest common divisor of $x^{r/2} \pm 1$ and n [7]. Table 1 shows some commonly used public key cryptographic algorithms that will be broken when Shor's algorithm can be easily implemented. Note that the security here is a measure of the approximate number of operations used by the best attacks against the cryptographic algorithm.

Algorithm	**Pre-quantum security**	**Post-quantum security**
DH-3072 Key Exchange	2^128	0
256 bit ECDH Key Exchange	2^128	0
RSA 3072 Signature	2^128	0
DSA 3072 Signature	2^128	0
RSA 3072 Encryption	2^128	0

Table 1: Algorithms broken by Shor's Algorithm

3.3.2 Grover’s Algorithm

Grover’s algorithm is a generic quantum search algorithm that involves searching through an unstructured large list of N items and finding a particular item that we want. It allows us to solve this problem which typically takes O(N) time in less than O($\sqrt{N}$) time – a quadratic speed up. Some simulations of Grover’s algorithm have been done on classical computers in the past, such as the one done by AB Mutiara and R Refianti [14], with conclusions supporting the quadratic speed up claims. The algorithm can also increase the efficiency of brute force attacks such as collision attacks and pre-image attacks, reducing the number of operations needed to break symmetric key schemes. The reduction of security by applying Grover’s algorithm is demonstrated in the table below.

Algorithm	**Pre-quantum security**	**Post-quantum security**
AES 128	2^128	2^64
AES 256	2^256	2^128
GMAC	2^128	2^128
SHA 256	2^256	2^128
SHA3 256	2^256	2^128

Table 2: Algorithms weakened by Grover’s Algorithm

4. Post quantum cryptographic schemes

There are several post quantum cryptographic schemes that have been theorized. These schemes “have solidly resisted every suggested attack” [1] and are strong because there is no way to apply Shor’s algorithm to these proposals. Thus they are considered to be plausible post-quantum cryptographic schemes that will likely be implemented in the future.

4.1 Lattice Based Cryptography

In mathematics, a Lattice is defined as a partially ordered set where every pair of points is regularly spaced out and the grid stretches out to infinity. To restrict a lattice into finite space so that it can be used for cryptographic encryption, the basis of a lattice is used. A basis is defined as a small collection of vectors that can be used to reproduce any point in the grid that forms the lattice. They are the n linearly independent set of vectors that generate the lattice itself.

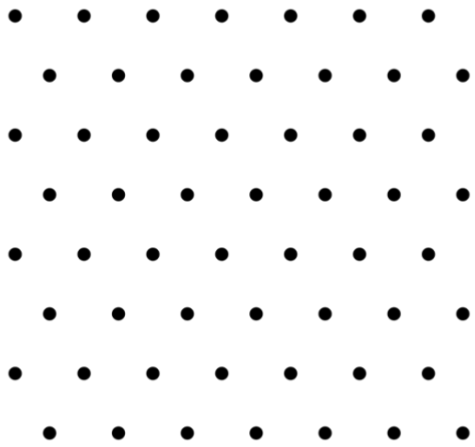

Figure 1: 2 Dimensional Lattice

Lattice based encryption schemes are based on hard mathematical problems related to lattices. The two main hard problems are the Short Vector Problem (SVP) and the Closest Vector Problem (CVP). The SVP involves outputting the shortest nonzero vector given a lattice. This problem is considered a NP (non-deterministic polynomial time) problem since the time complexity is approximately $2^{O(n)}$ using even the best algorithm (the LLL algorithm). The CVP is also a NP problem and is closely related to SVP, where, given a lattice and a target point, the lattice point closest to the target is being found. One of the main advantages of lattice systems is the worst-case security guarantee property, meaning that to break it, there would be a need to have a general algorithm that can solve lattice problems within polynomial time. This property also suggests that attacks on the algorithm are only successful for a small choice of parameters, meaning the construction and design of the scheme does not have fundamental flaws that compromise security. There has been no evidence of any quantum algorithms that perform significantly better at solving lattice problems than classical algorithms. NTRU is a public key cryptosystem that was proposed in 1996 and is one of the more prominent lattice-based systems. A general outline of how the system works is presented. The public key h is a p-coefficient polynomial with each coefficient in the set {0,1,…,q-1} and the ciphertext c is another polynomial in the same range. The sender can choose 2 secret polynomials d and e with small coefficients and compute

$$c = \big((hd + e) mod\ x^{p-1}\big)\ mod\ q.$$

Then the sender finds L where L is a set of pairs (u, v) of p-coefficient polynomials with integer coefficients such that the equation

$$0 = \big((hu - v) mod\ x^{p-1}\big)\ mod\ q$$

is true. If this condition is true, L is a lattice in 2p-dimensional space and it will contain a point close to (0, c), namely the point (d, c – e). If an attacker wants to find the secret polynomials d and e so they can try and break the ciphertext, given only ciphertext c and public key h, they will have to solve the CVP which is infeasible.

4.2 Code Based Cryptography

Code based cryptographic systems use an error correcting code as the underlying algorithmic primitive. They are based on coding theory and have several advantages over current public key schemes, such as lower algorithmic complexity and faster speeds. However, they are not currently used because of their main limitation: code based schemes have very large keys that can range from 100 kilobytes to several megabytes in size, requiring large amounts of memory to be viable. The most prominent public key encryption scheme is the McEliece public key cryptosystem which remains unbroken since its proposal in 1978. The key generation procedure for McEliece is highlighted below.

Step 1: Select binary (n,k) linear code C with decoding algorithm A
Step 2: Select a generator matrix G for C
Step 3: Select binary matrix S
Step 4: Select permutation matrix P
Step 5: Calculate public key ($\hat{G}$, t) and private key (S,P,A)

Table 3: Key generation in McEliece Public Key Cryptosystems

In step 1, the receiver selects a binary (n,k)-linear code C from some family of codes for which they know an efficient decoding algorithm A. This code is capable of efficiently correcting t errors and will usually be a binary Goppa code. The receiver can make the binary linear code public knowledge, but must keep the decoding algorithm a secret. In step 2, the receiver also selects any generator matrix G for binary code C, and it must be the case that knowing G would reveal the decoding algorithm A, so G must also be kept a secret. In step 3 and 4 respectively, the receiver selects two matrices, a binary matrix S and a permutation matrix P. Both matrices are kept a secret and are part of their private key. Finally in step 5, the public and private keys are calculated. For ease of notation, we denote $\hat{G}$ as S*G*P. A sender who wants to encode a message must first compute c' = m * $\hat{G}$. The sender must then generate a random vector e of length n, and this vector will contain exactly t one's. This random vector is noise that is added to the message. It is important that only t one's are added so that the code C can correct these errors during decoding. The sender can then send ciphertext c = c' + e, and the receiver can remove the random noise added with the decoding algorithm A and the binary code C and retrieve the original message by decoding c'. If there is an attacker who intercepts c, they cannot recover the message and remove the random noise since they do not know the factors of $\hat{G}$, the decryption algorithm A or the parameters n,k which specify the binary linear code C.

5. Critical Evaluation

Firstly, although Grover's algorithm speeds up the time it takes to brute force symmetric key algorithms, the quadratic speed up in time may not be as threatening as it seems. For example, AES 256 would still take, on average, 2^{128} operations to break. Some schemes that use shorter bit size keys (i.e. AES 128) will be compromised due to Grover's algorithm, however the general defense against it is to just have longer keys, for example, doubling the key length to maintain current levels of security. This also means that these algorithms are not completely broken by Grover's, so they will still be viable after quantum computing becomes prominent. Thus, the main emphasis in post quantum cryptography in my opinion should be to build schemes that can withstand Shor's algorithm rather than Grover's algorithm.

Many code-based systems are often restricted by the large memory requirements due to large key sizes. There have been several attempts to alter the original proposal made by McEliece to try and reduce the large public key [16], however, they have been unsuccessful, either due to a lack of security or efficiency. Moreover, McEliece's system when introduced in 1978 was generally overlooked because of this limitation, but since it is quantum resistant, it is a suitable candidate for post quantum cryptography. I believe that this suggests there are perhaps other cryptography algorithms out there that were overlooked due to limitations in our current era but are in fact suitable schemes for post quantum systems. Even if that isn't the case, as technology progresses and everything is upgraded, memory constraints will no longer be as large of a concern, thereby overcoming the limitation of code-based cryptosystems.

Lattice based mathematics are a branch of hard mathematical problems that have been studied for a long time and are well understood. This allows cryptographers to design schemes around lattice-based problems which can be easily understood and implemented by other cryptographers due to the large number of resources that can be used to understand them. Furthermore, the exponential complexity of both the SVP and CVP provide a strong mathematical basis for lattice based cryptographic systems. Moreover, they also have the advantage of having a stronger security guarantee as its construction is based on worse case hardness as opposed to other cryptosystems which are based on average case hardness, and this security is not achieved with key sizes that are too large to be deemed acceptable. There have been some potential attacks on lattice-based systems, however, NTRU remains unbroken. One attack uses the cyclotomic structure of $x^p - 1$ and combines it with an extension of Shor's algorithm. This attack did not affect the security of NTRU directly, however, variations of the attack could possibly be effective against NTRU if they are explored further. Overall, the advantages outweigh the limitations and make lattice-based systems the front runner for post quantum crypto schemes, and, in my opinion, I believe they are the most promising for cryptosecurity schemes in the post-quantum era.

6. Conclusion & Future Direction

At this moment in time, it seems that the literature in this field is limited, and many studies have not presented concrete conclusions with solid evidence to suggest which schemes will be the most suitable in the post quantum era. There is a need for more research and testing to determine whether the proposed cryptographic algorithms can withstand every kind of attack that quantum computers could make. Furthermore, more research into other forms of post quantum cryptographic systems need to be explored, such as multivariate based systems that use multivariate quadratic polynomial maps as their one-way trap door function [15]. These other systems could potentially be just as strong if not stronger than lattice systems and could meet our security needs for decades to come. With so many possibilities, the future of post-quantum cryptography is incredibly exciting.

7. References

[1] DJ. Bernstein, T. Lange. “Post-quantum cryptography” in Nature, vol. 549, pp. 188-194, 2017.

[2] NY. Goshwe. “Data Encryption and Decryption Using RSA Algorithm in a Network Environment” in IJCSNS International Journal of Computer Science and Network Security, vol. 13, no. 7, pp. 8-13, 2013.

[3] UM. Maurer, S. Wolf. “The Diffie-Hellman Protocol” in Designs, Codes and Cryptography, vol. 19, pp. 147-171, 2000.

[4] TD. Ladd, F. Jelezko, R. Laflamme, Y. Nakamura, C. Monnroe, JL. O’Brien. “Quantum Computers” in Nature, vol. 464, pp. 45-53, 2010.

[5] OH. Montiel Ross, "A Review of Quantum-Inspired Metaheuristics: Going From Classical Computers to Real Quantum Computers" in IEEE Access, vol. 8, pp. 814-838, 2020.

[6] KA. Valiev. “Quantum computers and quantum computations” in Physics-Uspekhi, vol. 48, pp.1, 2005.

[7] M. Hayward. “Quantum Computing and Shor’s Algorithm”. Sydney: Macquarie University Mathematics, 2008.

[8] MJ. Nene, G. Upadhyay. “Shor’s Algorithm for Quantum Factoring” in Advanced Computing and Communication Technologies. Advances in Intelligent Systems and Computing, vol. 452, pp. 325-331, 2016.

[9] SMS. Hussain, SM. Farooq, TS. Ustun, "Analysis and Implementation of Message Authentication Code (MAC) Algorithms for GOOSE Message Security" in IEEE Access, vol. 7, pp. 80980-80984, 2019, doi: 10.1109/ACCESS.2019.2923728.

[10] KA. Sangeeta. “A Review on Symmetric Key Cryptography Algorithms” in International Journal of Advanced Research in Computer Science, vol.8 (4), pp. 358-361, 2017.

[11] H. Nejatollahi, N. Dutt, S. Ray, F. Regazzoni, I. Banerjee, R. Cammarota. “Post-Quantum Lattice-Based Cryptography Implementations: A Survey” in ACM Computing Surveys, vol.51 (6), pp. 1-41, 2019.

[12] Sendrier N. “Code-Based Cryptography” in van Tilborg H.C.A., Jajodia S. (eds) Encyclopedia of Cryptography and Security. Springer, Boston, MA, 2011, https://doi.org/10.1007/978-1-4419-5906-5_378

[13] AW. Mohsen, AM. Bahaa-Eldin, MA. Sobh, "Lattice-based cryptography," in *2017 12th International Conference on Computer Engineering and Systems (ICCES)*, pp. 462-467, 2017, doi: 10.1109/ICCES.2017.8275352.

[14] AB. Mutiara, R. Refianti. “Simulation of Grover’s Algorithm Qunatum Search in a Classical Computer” in International Journal of Computer Science and Information Security, vol.8 (9), pp.1-9, 2010.

[15] J. Ding, A. Petzoldt, "Current State of Multivariate Cryptography," in IEEE Security & Privacy, vol. 15 (4), pp. 28-36, 2017, doi: 10.1109/MSP.2017.3151328.

[16] Overbeck, Raphael & Sendrier, Nicolas. (2009). Code-Based Cryptography. Doi: 10.1007/978-3-540-88702-7_4.

8. Self-Reflection

Researching this topic on post-quantum cryptography was very stimulating. I felt that I learnt a lot about how cryptography will be used in the post-quantum era. It made me realize that even though it is not something that will likely affect us for a while, there is still an urgent need to explore and test more secure systems. I also found the proposed quantum systems very interesting and will continue to keep up to date with the latest in this field. Overall, I have really enjoyed working on this project and I hope to explore this topic more in the future.